ACOUSTIC EVENT DETECTION USING SPEAKER RECOGNITION TECHNIQUES: MODEL OPTIMIZATION AND EXPLAINABLE FEATURES

Mattson Ogg *Benjamin Skerritt-Davis*

Johns Hopkins University Applied Physics Laboratory
11100 Johns Hopkins Road, Laurel, MD 20723 USA
{mattson.ogg, ben.skerritt-davis}@jhuapl.edu

ABSTRACT

We adapted methods from the speaker recognition literature to acoustic event detection (or audio-tagging) and applied representational similarity analysis, a cognitive neuroscience technique, to gain a deeper understanding of model performance. Experiments with a feed-forward time-delay neural network architecture (TDNN) were carried out using the FSDKaggle2018 dataset. We examined various system optimizations such as speed and reverb augmentation, different input features (spectrograms, mel-filterbanks, MFCCs and cochleagrams), as well as updates to the network architecture (increased or decreased temporal context and model capacity as well as drop-out and batch-normalization). Most system configurations were able to outperform the original published baseline and, primarily using speed augmentation, our system was able to outperform a harder baseline derived from a model pre-trained on many times more data. Additional experiments applying representational similarity analysis to the network embeddings allowed us to understand what acoustic features the different systems used to perform the task.

Index Terms— Audio tagging, acoustic event recognition, speaker recognition, explainable features, acoustic features

1. INTRODUCTION

Objects and events in the environment can be recognized based on the sound patterns they generate [1]. Automated sound identification (also called sound/acoustic event recognition or audio tagging) supports many critical information retrieval [2], hearing assistance [3], urban planning [4], and monitoring [5] applications. Recognition systems can also provide insight into human [6] and animal [7] auditory processing. Despite its obvious importance, sound recognition is not a solved problem. Continued development will help address many remaining gaps in performance and yield new insights for acoustics and machine perception research.

Many modern approaches to sound recognition take the rich deep learning literature on visual object recognition as a starting place [8, 9, 10]. These methods essentially treat the task as an image classification problem based on a spectrogram input. However, such an approach has a number of shortcomings (see [11] for discussion). Most notably it is a sub-optimal treatment of the inherently temporal nature of sound. That is, to comply with the constraints of image recognition methods, an incoming audio example is normally chunked into fixed size spectrogram images that are fed to a 2-dimensional convolutional network architecture. Chunks are then classified individually or the systems' hypotheses are averaged over time points. This results in a slightly awkward treatment of audio examples that frequently vary in duration. Also, if examples are labelled at the file-level, individual chunks run the risk of not containing any information related to the target sound due to pauses or interruptions. Indeed, some results have suggested that chunk averaging may be sub-optimal relative to systems designed to handle variable length input [12].

Other lines of research have emerged for identifying specific classes of sound sources such as human speakers [13, 14]. Speaker recognition in particular has made remarkable progress using deep learning techniques that stem from language and speech modeling [15, 16]. These approaches give special attention to how information unfolds in time [17], for example, by statistically pooling time windows to create an intermediate global representation within the network [18] that is further processed. This approach can also be used to recognize acoustic scenes [19].

With these branches of acoustic research in mind we present a series of experiments where we applied successful techniques from speaker identification to the task of sound event recognition. We also study adaptations of this framework to better fit the sound event recognition task. In the end, we achieved accurate performance on a well-studied recognition dataset [20], beating the initial published baseline as well as another stronger baseline that uses an image-recognition-based system pre-trained on many more hours of data [8]. Finally, we provide insight into the performance differences among these systems by exploring their respective embedding spaces using representational similarity analysis [21].

2. STUDY DESIGN

2.1. Data and Task

Our goal was to apply speaker recognition techniques to sound event recognition. Thus, we aimed to select a dataset that was organized analogously to typical speaker recognition datasets [14, 22]. This involves a large number of discrete target classes with numerous training examples where each example corresponds to a single target class. We chose to conduct our experiments using the popular audio-tagging dataset, FSDKaggle2018 (see [20] for details). Briefly, this is a curated collection of publicly available data that is user generated and user tagged. The data was then binned by the authors of the corpus into 41 discrete classes

according to a well-established acoustic event ontology [2]. Clips vary in duration and the number of training examples is unbalanced across classes. Our experiments were carried out on the full (manual and user-labelled) portion of the dataset. This dataset does not contain a validation partition, so we created one by holding out one third of the manually labelled examples of each class from the training partition. Model selection was done on this validation set. We then report final system performance of the selected model on the test partition. All audio data were down-sampled to a 16kHz sampling rate.

We selected this dataset because, relative to other options, it strikes a good balance in terms of dataset size (up to 18 hours of training data), diversity (41 discrete classes), and label specificity (generally, a single class-label per clip). Some more thoroughly labelled datasets exist but are limited in their number of target classes [23] or training examples [24]. While other larger datasets often sacrifice label purity and usually contain multiple classes per clip [2, 4, 9].

We compare performance against the published baseline for this dataset from [20] (mAP@3 = 0.70). A harder baseline was also generated using Google's YAMNet system trained on their AudioSet corpus [8, 25]. We extracted YAMNet embeddings for each sound token (no augmentation) and then trained a single fully connected layer between those embeddings and 41 output units for the corpus' target classes. The shallow YAMNet embedding network was trained for 100 epochs and we retained the model with the highest performance on the validation partition (validation performance: accuracy = 0.79, mAP@3 = 0.86; test performance: accuracy = 0.78, mAP@3 = 0.85).

2.2. Network Architecture and Training

Our initial TDNN model was an implementation of the x-vector system developed by Snyder and colleagues for speaker recognition [18, 26] that we reproduced in PyTorch (using [27]). Because it is based on feed-forward units, TDNN networks are faster and more efficient to train than recurrent networks, such as LSTMs (c.f., [17]).

A TDNN models temporal context via a hierarchy of layers that progressively see larger windows of time via dilations that occur in higher layers. Variable length audio input is handled by a combination of frame-level and segment-level components within the model. At the frame-level, the TDNN structure slides over frames of the variable length input. The output of these layers is projected to a set of units over which the mean and standard deviation are calculated for a given example (audio file). The mean and standard deviation of these units are concatenated to comprise a statistics pooling layer that begins the segment-level processing. Above the statistics pooling layer are two fully connected layers (which comprise the embedding layers) followed by 41 output units, one for each of the target classes.

Specifically, we re-created the architecture described by [26] with a context of five input time-frames (of the input spectrogram) in the first layer. Layer two received layer one's output with a context of three frames and a dilation of two before sending output to layer three which also has a context of three frames with a dilation of three. Layers four and five both have contexts of one. Thus, layers three and higher operated over a total context of 15 spectrogram frames. All TDNN and embedding layers have 512 units. The layer just prior to the statistics pooling layer projected to 1500 units that were used to calculate a mean and standard deviation which were concatenated before output to the first fully connected layer, followed by the second fully connected layer (each with 512 units).

Audio files were represented to the network via a time-frequency representation. Speaker and sound event recognition studies have used a variety of different inputs, so we tested multiple popular representations to determine an optimal set of features. We explored many Kaldi-style representations using torchaudio all with a 25-ms window and 10-ms hop size. Because we used a higher sample rate (16k) than the original x-vector network implementation (8k), we allowed an increase in the number of frequency bins for each representation: spectrograms (201 frequency bins), mel-filterbanks (80 mel bins) and MFCCs (mel-frequency cepstral coefficients; 40 cepstral features). We also generated a cochleagram representation to approximate the peripheral auditory system of a human listener (which we instantiated via [28]; upper frequency limit = 8k, 4 times overcomplete bandpass filter sampling, output down sampled to 100 Hz). All audio was zero-padded for 5-ms at onset and up to 5 ms at offset before windowing. During training, each input spectrogram was normalized (between 0 and 1), and the durations of input examples were standardized to between 1-second and 30 seconds either via looping spectrograms that were too short (until they exceeded 101 frames), or by truncating them (to 3001 frames if they exceeded that).

Each training run comprised 100 epochs and we retained the model from the epoch with the highest accuracy on the

Table 1: Performance of different model architectures and training configurations on the validation partition. Parentheses indicate change in performance from our baseline TDNN model on the first line.

	Accuracy				mAP@3			
	Cochleagram	Mel-Filterbank	MFCC	Spectrogram	Cochleagram	Mel-Filterbank	MFCC	Spectrogram
Initial Baseline TDNN Model	0.73(Δ0)	0.74(Δ0)	0.24(Δ0)	0.75(Δ0)	0.79(Δ0)	0.8(Δ0)	0.32(Δ0)	0.81(Δ0)
Diff. Maps	0.74(Δ0.016)	0.75(Δ0.012)	0.16(Δ-0.078)	0.75(Δ0.002)	0.8(Δ0.006)	0.81(Δ0.007)	0.24(Δ-0.087)	0.81(Δ-0.005)
Reverb Aug.	0.77(Δ0.043)	0.79(Δ0.043)	0.7(Δ0.464)	0.77(Δ0.022)	0.82(Δ0.03)	0.84(Δ0.035)	0.78(Δ0.453)	0.82(Δ0.011)
Speed Aug.	0.83(Δ0.099)	0.88(Δ0.134)	0.76(Δ0.526)	0.87(Δ0.117)	0.87(Δ0.079)	0.91(Δ0.103)	0.82(Δ0.493)	0.9(Δ0.089)
Smaller Net: 256 Units	0.72(Δ-0.004)	0.76(Δ0.019)	0.45(Δ0.218)	0.74(Δ-0.01)	0.78(Δ-0.012)	0.82(Δ0.013)	0.56(Δ0.234)	0.8(Δ-0.011)
Larger Net: 1024 Units	0.74(Δ0.015)	0.75(Δ0.007)	0.4(Δ0.166)	0.76(Δ0.007)	0.8(Δ0.009)	0.81(Δ0.007)	0.5(Δ0.178)	0.81(Δ0.001)
Reduced Context-Layer	0.72(Δ-0.011)	0.74(Δ-0.002)	0.43(Δ0.194)	0.73(Δ-0.02)	0.78(Δ-0.018)	0.81(Δ0.002)	0.54(Δ0.216)	0.8(Δ-0.015)
Added Context-Layer	0.74(Δ0.007)	0.75(Δ0.011)	0.56(Δ0.326)	0.74(Δ-0.012)	0.79(Δ-0.001)	0.81(Δ0.001)	0.65(Δ0.332)	0.8(Δ-0.015)
Batch-Norm, Drop-Out	0.76(Δ0.031)	0.8(Δ0.055)	0.63(Δ0.39)	0.78(Δ0.033)	0.81(Δ0.016)	0.85(Δ0.044)	0.71(Δ0.392)	0.84(Δ0.026)
Speed+Reverb Aug.	NA	0.87(Δ0.126)	NA	NA	NA	0.9(Δ0.093)	NA	NA
Speed+Reverb, Diff. Maps, Batch-Norm+Drop-Out	NA	0.86(Δ0.113)	NA	NA	NA	0.89(Δ0.084)	NA	NA
Speed+Reverb, Batch-Norm+Drop-Out	NA	0.85(Δ0.109)	NA	NA	NA	0.89(Δ0.084)	NA	NA
Speed+Reverb, Diff. Maps	NA	0.87(Δ0.129)	NA	NA	NA	0.91(Δ0.1)	NA	NA

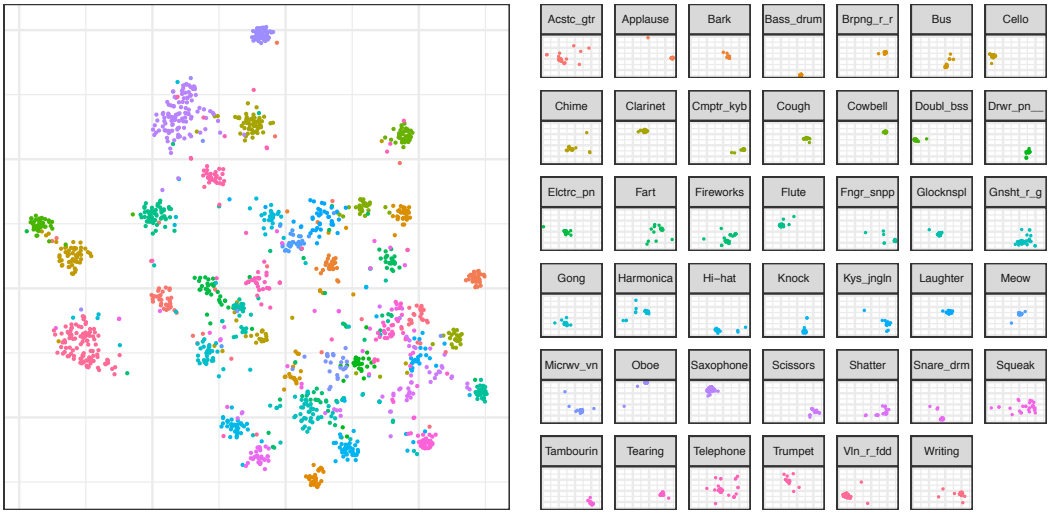

Figure 1: Our top performing TDNN model's embedding space visualized using t-SNE. Points correspond to examples in the test set. Insets in the right panel call out examples for each target class and serves as a color code.

validation partition. Batches were of 16 examples (shuffled between epochs). We used PyTorch's cross-entropy loss function with a stochastic gradient descent optimizer (learning rate: 0.001, momentum: 0.9, weight decay: 0.001). Because there was an imbalance of training data among classes, weights were applied in the loss function that gave more weight to low occurrence classes relative to the class with the highest number of training examples. We report raw accuracy on the validation and test partitions as well as mean Average Precision @ 3 (mAP@3).

2.3. Study of Network Architecture and Training Parameters

We carried out a number of experiments to optimize inputs and model parameters during training. The baseline for these experiments was the performance of a network structured like the original x-vector network configuration [26], albeit with a larger input representation to take advantage of the higher sampling rate (see above). Throughout, we compared the performance of four front-end, time-frequency representations (spectrogram, MFCC, mel-filterbank, or cochleagram input). In terms of input optimizations we also attempted to help the network efficiently learn spectral and temporal variability cues by appending 2 "difference maps" to the input time-frequency representation: 1) the first derivative in each frequency channel over time and, 2) first derivative in each time bin over frequencies.

We then examined the effectiveness of common data augmentation strategies gleaned from work on speech tasks [29, 30] (see also [31]): simple speed augmentation (plus and minus 10%, thus altering any pitch by the same amount) and reverb augmentation (instantiated via [32]), both of which were carried out on the audio files prior to extracting a time-frequency representation. Augmentation by background noise and babble was not examined, since these recordings already contained some. Sounds similar to those in this dataset are also often used in noise augmentation for speech tasks, which risked confusing class labels during training.

Next, we turned our attention to optimizing the network architecture in various ways: varying the number of units in each layer (feed forward layer 256 or 1024 and number of statistics pooling layer units 750 and 3000, respectively), increasing or decreasing temporal context before statistical pooling (by removing or duplicating layer 3), and adding batch norm and 50% dropout. Based on these experiments we studied a final set of models that used combinations of the best performing parameters and optimizations.

3. RESULTS

Model performance is summarized within Table 1. Differences relative to our baseline TDNN configuration are indicated in parentheses. Without any modifications the initial TDNN models outperformed the original published baseline given most of the feature input options, although performance using MFCCs was generally poor. Not every optimization we experimented with improved performance and some optimizations varied in how much they improved performance given different input features.

Table 2: Description of some acoustic features used in the representational similarity analyses.

Feature	Description	Interpretation
Log-Attack-Time[1]	Log of the time difference between attack onset and ending	Lower values = faster onset time
Temporal Centroid[1]	Center of gravity of the energy envelope	Lower values = earlier temporal centroid
Spectral Centroid[2]	Center of gravity of the spectral (ERB) envelope	High values = higher frequency centroid
Spectral Flatness[2]	Ratio of geometric and arithmetic means of the ERB spectra	Measures noise/harmonic content. Higher values are flatter/noisier
Spectral Variability[2]	1 minus the correlation of ERB channel spectra between timepoints	Higher values = more variable envelope
Aperiodicity[3]	Amount of aperiodic energy in the signal	Higher values = more aperiodic
ERB Energy[2]	Amount of energy in the spectral representation at each timestep	Sum of squared amplitudes in the spectral representation.
Raw ERB cochleagram[2]	Raw ERB representation of each of 77 channels: 30 Hz to 16 kHz	Energy in each channel over time.
Mod. Power Spectrum[4]	2D-FFT of Gaussian spectrogram	Degree of joint spectral/temporal modulation rates

Derived based on: 1) Energy envelope or, 2) ERB (cochleagram) representation in the Timbre Toolbox [35, 36], 3) YIN [37], 4) modulation power spectrum [38]. Table adapted from [33].

Speed and, to a lesser degree, reverb augmentation were particularly beneficial, as was batch-norm and drop-out. Changing temporal context and model capacity often hurt performance.

Only with speed augmentation were some of our TDNN models able to beat the performance of the stronger YAMNet-baseline system on the validation partition. To further improve our system we explored combinations of optimizations that were beneficial in the initial experiments. These combination experiments were carried out using mel-filterbank input features because these achieved the highest performance relative to other initial TDNN systems. A model trained with speed and reverb augmentation with the spectrotemporal difference maps using the mel-filterbanks as input features achieved the highest validation performance among these combination experiments. However, no combination experiments out-performed the initial mel-filterbank TDNN model trained with speed-augmented data, so this was selected as our final model to evaluate the test data (accuracy = 0.82, mAP@3 = 0.86), which slightly out-performed the YAMNet-baseline. Class separation within the embedding space of this top-performing model is visualized in Figure 1.

4. REPRESENTATIONAL SIMILARITY ANALYSIS

Performance of our final model and the YAMNet-baseline were both quite high, despite operating over the audio differently. Thus, we were interested in better understanding whether any differences existed in how these systems internally represented audio examples and the influence of different acoustic qualities. To do this, we employed a method called representational similarity analysis [21] which can provide a high-level understanding of complex systems by correlating inter-item distances among different representations of a set of probe examples. We extracted network embeddings (i.e., activations from the layer just prior to the 41-class output layer) from our final, best performing model and from the YAMNet model for each example in the test partition. Then among each model's embeddings, we calculated the cosine distance of the network representations for each pair of test examples, to populate two 1600 by 1600 network-dissimilarity matrices (one matrix for each network). The network-dissimilarity matrices were compared against another set of inter-item, acoustic-dissimilarity matrices (absolute value of feature differences) for a set of well-studied acoustic features derived for each test item (see Table 2 and [33] for detailed description). These acoustic distances were contained within another set of 1600 by 1600 acoustic-dissimilarity matrices, (one matrix per feature). Note, because these dissimilarity matrices are symmetrical across the diagonal, only one unique item pairing was analyzed (e.g., item-1 vs item-2 or item-2 vs item-1).

We carried out rank-order semi-partial Spearman correlations between each network-dissimilarity matrix and the set of acoustic-dissimilarity matrices. In each semi-partial test, a correlation was derived between the network-dissimilarity matrix, and the target acoustic-dissimilarity matrix, while holding the other features constant. Only correlations that were interpretable (i.e., positive) and statistically significant after false-discovery rate correction were retained.

The representational similarity analysis is summarized in Figure 2. We found that despite their high performance, our model and the YAMNet model's embedding spaces were only modestly correlated (r_s = 0.31). Both models' performance was

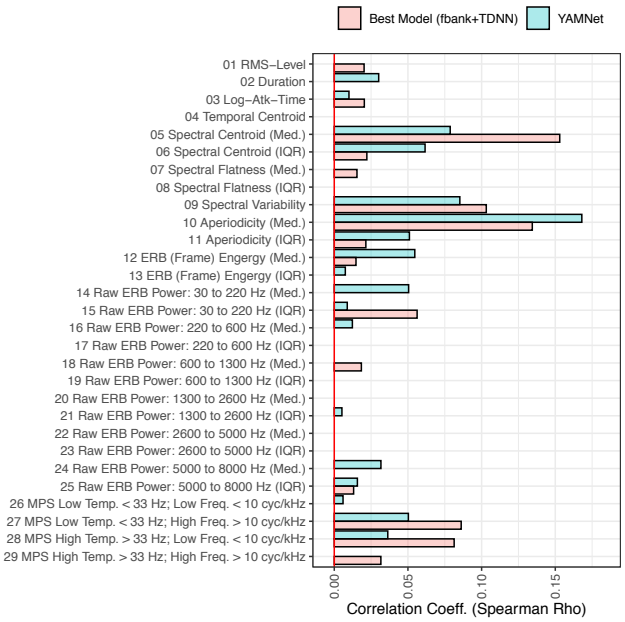

Figure 2: Representational similarity analysis of both the top performing TDNN model and the YAMNet embeddings.

most strongly associated with acoustic cues for aperiodicity, spectral centroid, and spectral variability, albeit with differences in the relative importance of these features. This is similar to features that influence dissimilarity ratings among human listeners [33] and neural representations [34].

5. CONCLUSION

We examined the effectiveness of different speaker recognition methods on an audio-tagging task. We were able to obtain good performance, beating the original baseline for this dataset, and a more challenging YAMNet-baseline derived from a system trained on many hours more data. The TDNN architecture appears to derive great performance benefit from data augmentation (particularly speed augmentation). Representational similarity analyses implicated a set of acoustic features that are also associated with sound recognition in the human auditory system.

6. REFERENCES

[1] F. E. Theunissen, and J. E. Elie, "Neural processing of natural sounds," *Nat. Rev. Neurosci.*, vol. 15, pp. 355–366, 2014.

[2] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, ... and Ritter, M. "Audio set: An ontology and human-labeled dataset for audio events," In *Proc. IEEE ICASSP*, 2017, pp. 776-780.

[3] A. Hüwel, K. Adiloğl, and J. H. Bach, "Hearing aid research data set for acoustic environment recognition." In *Proc. IEEE ICASSP,* 2020, pp. 706-710.

[4] M. Cartwright, A. E. M. Mendez, J. Cramer, V. Lostanlen, G. Dove, H.-H. Wu, J. Salamon, O. Nov, and J. Bello, "SONYC urban sound tagging (SONYC-UST): A multilabel dataset from an urban acoustic sensor network," in *Proc. IEEE DCASE*, 2019, pp. 35–39.

[5] D. Stowell, T. Petrusková, M. Šálek, and P. Linhart, "Automatic acoustic identification of individuals in multiple species: improving identification across recording conditions," *Journal of The Royal Society Interface*, vol. 16, 20180940, 2019.

[6] A. Kell, D. Yamins, E. N. Shook, S. Norman-haignere, and J. H. McDermott, "A task-optimized neural network replicates human auditory behavior, predicts brain responses, and reveals a cortical processing hierarchy," *Neuron*, vol. 98, 2018.

[7] J. E. Elie, and F. E. Theunissen, "Zebra finches identify individuals using vocal signatures unique to each call type," *Nat. Comm.*, vol. 9, pp. 1-11, 2018.

[8] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J.F. Gemmeke, A. Jansen, C. Moore, M. Plakal, D. Platt, R.A. Saurous, B. Seybold, M. Slaney, R. Weiss, and K. Wilson, "CNN architectures for large-scale audio classification," in *IEEE ICASSP*, 2017, pp. 131–135.

[9] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "FSD50K: An open dataset of human-labeled sound events," *arXiv preprint arXiv:2010.00475*, 2020.

[10] A. Guzhov, F. Raue, J. Hees, and A. Dengel, "ESResNet: Environmental sound classification based on visual domain models," *arXiv preprint arXiv:2004.07301*, 2020.

[11] E. M. Kaya and M. Elhilali, "Modelling auditory attention," *Philosophical transactions of the Royal Society of London. Series B, Biological Sciences*, vol. 372, no. 1714, p. 20160101, 2017.

[12] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: A large-scale speaker identification dataset," in *Proc. Interspeech*, 2017, pp. 2616–2620.

[13] J.H.L. Hansen and T. Hasan, "Speaker recognition by machines and humans: A tutorial review," *IEEE Signal Proc. Mag.*, vol. 32, no. 6, pp. 74–99, 2015.

[14] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, "Voxceleb: Large-scale speaker verification in the wild," *Computer Speech & Language*, vol. 60, 101027, 2020.

[15] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Proc. Interspeech*, 2015.

[16] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang, "Phoneme recognition using time-delay neural networks," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 3, pp. 328–339, Mar. 1989.

[17] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in *Proc. IEEE ICASSP*, 2016, pp. 5115–5119.

[18] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in *Proc. Interspeech*, pp. 999–1003, 2017.

[19] A. Jati, A. Nadarajan, K. Mundnich, and S. Narayanan "Characterizing dynamically varying acoustic scenes from egocentric audio recordings in workplace setting," *arXiv preprint arXiv:1911.03843*, 2019.

[20] E. Fonseca, M. Plakal, F. Font, D. P. W. Ellis, X. Favory, J. Pons, and X. Serra, "General-purpose tagging of freesound audio with audioset labels: Task description, dataset, and baseline," in *Proc. IEEE DCASE*, 2018, pp. 69–73.

[21] N. Kriegeskorte and R. A. Kievit, ''Representational geometry: Integrating cognition, computation, and the brain,'' *Trends Cognit. Sci.*, vol. 17, no. 8, pp. 401–412, Aug. 2013.

[22] M. McLaren, L. Ferrer, D. Castan, and A. Lawson, "The speakers in the wild (SITW) speaker recognition database," in *Proc. Interspeech*, 2016, pp. 818–822.

[23] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *22nd ACM International Conference on Multimedia (ACM-MM'14)*, 2014, pp. 1041–1044.

[24] K. J. Piczak, "Environmental sound classification with convolutional neural networks," in *25th International Workshop on Machine Learning for Signal Processing (MLSP)*, 2015, pp. 1–6.

[25] https://github.com/tensorflow/models/tree/master/research/audioset/yamnet

[26] D. Snyder, D. Garcia-Romero, G. Sell, et al., "X-vectors: Robust DNN embeddings for speaker recognition," in *Proc. IEEE ICASSP*, 2018, pp. 5329–5333.

[27] https://github.com/cvqluu/TDNN

[28] https://pycochleagram.readthedocs.io/en/latest/index.html

[29] T. Ko, V. Peddinti, D. Povey, M. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on. IEEE, 2017, pp. 5220–5224.

[30] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Proc. INTERSPEECH*, 2015, pp. 3586–3589.

[31] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 279–283, 2017.

[32] https://github.com/mravanelli/pySpeechRev

[33] M. Ogg, and R. L. Slevc, "Acoustic correlates of auditory object and event perception: Speakers, musical timbres, and environmental sounds," *Front. Psychol.*, vol. 10, 1594, 2019.

[34] M. Ogg, T. A. Carlson, and R. L. Slevc, "The rapid emergence of auditory object representations in cortex reflect central acoustic attributes," *J. Cogn. Neurosci.*, vol. 32, pp. 111–123.

[35] G. Peeters, B. L. Giordano, P. Susini, N. Misdariis, and S. McAdams, "The timbre toolbox: Extracting audio descriptors from musical signal," *J. Acoust. Soc. Am.*, vol. 130, no. 5, pp. 2902–2916, 2011.

[36] S. Kazazis, E. Nicholas, P. Depalle, and S. McAdams, "A performance evaluation of the timbre toolbox and the mirtoolbox on calibrated test sounds," in *Proc. of the Int. Symposium on Musical Acoustics (ISMA)*, 2017, pp. 144–147.

[37] A. De Cheveigné, and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music," *J. Acoust. Soc. Am.*, vol. 111, pp. 1917–1930, 2002.

[38] T. M. Elliott and F. E. Theunissen, "The modulation transfer function for speech iintelligibility," *PLoS Comput. Biol.*, vol. 5, no. 3, pp. 1–14, Mar. 2009.